



**SAVONIA**

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIIKAN JA LIIKENTEEN ALA

# HUBOT-OHJELMISTOROBOTIN KEHITYS SLACK-CHAT-SOVELLUKSEEN

TEKIJÄ/T: Jere Tuppurainen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Jere Tuppurainen	
Työn nimi Hubot-ohjelmistorobotin kehitys Slack-chatsovellukseen	
Päiväys 8.9.2019	Sivumäärä/Liitteet 23
Ohjaaja(t) Jussi Koistinen, Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) Enfo Oyj	
<p>Tiivistelmä</p> <p>Opinnäytetyön aiheena oli kehittää Slack-chatsovellukseen Hubot-ohjelmistorobotti, minkä tarkoituksena on hakea ja muokata ServiceNow palvelussa olevia tikettejä.</p> <p>Työ on toteutettu käyttäen Node.js ohjelmointikieltä ja se on open source projekti MIT:n lisenssillä. Hubottia pystyy kehittämään myös CoffeeScript kielellä, mutta kehityskieleksi valikoitui Node.js sen ollessa hieman tutumpi ja helpompi kuin CoffeeScript. Hubotin ideana on, että asennettassa siinä on jo tiettyjä skriptejä ja toimintoja valmiina asennettuna ja kehittäjät tekevät omia luokkakirjastoja ja funktioita omien tarpeidensa mukaan. Tässä työssä käytettiin ulkopuolista servicenow-lite nimistä npm-pakettia, minkä avulla voi helposti tehdä SOAP-pyyntöjä ServiceNow-instanssiin. Kaikkia tarvittavia metodeja ei kuitenkaan ollut servicenow-lite:ssa, vaan ne piti itse luoda sinne.</p> <p>Lopputuloksena syntyi Slackissa toimiva sovellus, minkä avulla pystyy hakemaan tietoja ServiceNow:n tiketeistä, ohjaamaan tikettejä toisille henkilölle tai ryhmälle, sulkemaan tikettejä, kommentoimaan tikettejä ja hakemaan kommentteja tiketeistä.</p>	
Avainsanat Node.js, Hubot, ServiceNow, Slack, SOAP API	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Jere Tuppurainen			
Title of Thesis Developing Hubot Chatbot to Slack Chat Application			
Date	September 8, 2019	Pages/Appendices	23
Supervisor(s) Mr Jussi Koistinen, Senior Lecturer, Mr Keijo Kuosmanen, Senior Lecturer			
Client Organisation /Partners Enfo Oyj			
<p><b>Abstract</b></p> <p>The subject of this thesis was to develop the Hubot chatbot to the Slack Chat application. The goal of the chatbot is to get and update tickets from ServiceNow.</p> <p>The chatbot was developed using the Node.js programming language and the chatbot is an open source project with MIT License. Hubot can also be developed with the CoffeeScript programming language but Node.js was chosen as primary developing language because it was already familiar to the developer and easy to use. The idea of Hubot is that it has some core scripts already installed when it is used for the first time and developers then write their own scripts in Hubot. This chatbot uses a third-party library called servicenow-lite which is used to make SOAP API calls to ServiceNow instance. Some necessary methods were missing in servicenow-lite so they had to be created.</p> <p>The result of thesis was a working chat application which can get data about tickets in ServiceNow, assign tickets to other persons or groups, close tickets, give comments to tickets and get comments from tickets.</p>			
<p><b>Keywords</b> Node.js, Hubot, ServiceNow, Slack, SOAP API</p>			

## ESIPUHE

Haluan kiittää Enfo Oyj:n väkeä mielenkiintoisesta opinnäytetyön aiheesta. Halusin, että opinnäytetyön aihe olisi mielenkiintoinen ja sitä tehdessä oppisi jotain uutta. Nämä kohdat täyttyivät hyvin.

Kuopiossa 8.9.2019 Jere Tuppurainen

## SISÄLTÖ

Termit ja lyhenteet .....	6
1 JOHDANTO .....	7
1.1 Toteutustavan valinta .....	7
1.2 Toimeksiantaja .....	8
2 KEHITYSKOhteet JA TYÖKALUT .....	9
2.1 Hubot .....	9
2.1.1 Hubotin scriptit .....	9
2.2 Git & GitHub .....	11
2.3 MIT – lisenssi .....	11
2.4 Node.js .....	11
2.4.1 Node.js:n toiminta .....	12
2.5 ServiceNow .....	12
2.5.1 ServiceNow:n Taskit .....	13
2.6 Slack .....	13
2.7 SOAP API .....	14
2.8 Visual Studio Code .....	14
3 KEHITYSTYÖN VAIHEET .....	15
3.1 Hubotin asennus .....	15
3.2 Slack adapterin käyttöönotto .....	17
3.3 Ominaisuuksien luominen .....	17
3.3.1 Tiketin ohjaaminen .....	18
3.3.2 Tiketin kommentointi .....	18
3.3.3 Tiketin sulkeminen .....	19
3.3.4 Tiketin kommenttien näyttäminen .....	19
3.3.5 Tiketin talletus muistiin .....	20
4 LOPPUTULOS .....	22
5 LÄHDELUETTELO .....	23

## Termit ja lyhenteet

Hubot	GitHubin luoma ohjelmistorobotti, jota voidaan käyttää erilaisten chat-sovellusten kanssa.
Git	Ilmainen hajautettu versionhallintajärjestelmä, joka on suunniteltu erilaisille ohjelmistoprojekteille.
GitHub	Web-pohjainen säilytyspaikka Gitin repositoryille.
Node.js	Avoimen lähdekoodin, kaikille alustoille sopiva JavaScript ajoympäristö, mitä käytetään palvelimen pään skriptien kirjoittamiseen.
MIT-lisenssi	Vapaa lisenssi ohjelmistoille, joka antaa käyttäjälle vapaasti käyttää, kopioida ja muokata teosta.
ServiceNow	Pilvipohjainen SaaS palvelunhallintajärjestelmän tuottaja, jolla tuotetaan yrityksille mm. help desk palveluita.
SOAP API	Simple Object Access Protocol, XML pohjainen viestintä protokolla.
Task/Tiketti	ServiceNow:n tapahtuma, esimerkiksi Incident tai Problem, joita esimerkiksi ServiceDesk käsittelee.
Visual Studio Code	Microsoftin kehittämä ilmainen tekstieditori ohjelmistokehitykseen.

## 1 JOHDANTO

Opinnäytetyön tarkoituksena oli kehittää Hubot-ohjelmistorobotti Slack-chatsovellukseen. Sovelluksen tarkoituksena on, että käyttäjä kykenee Slack-chatsovelluksen kautta käsittelemään ServiceNow:n tikettejä ja sen tarkoitus on helpottaa käyttäjää yksinkertaisten ja toistuvien toimintojen kanssa.

Työn määrittelyvaiheessa valittiin neljä tärkeintä ominaisuutta, mitkä ohjelmistorobotille kehitettiin opinnäytetyön puitteissa. Nämä ominaisuudet olivat tiketin ohjaaminen toiselle henkilölle tai ryhmälle, yksittäisen tiketin kommentointi, tiketin sulkeminen, sekä tiketin kommenttien hakeminen. Ohjelmistorobotille luotiin myös ominaisuus, millä pystyy hakemaan haluamansa tiketin tietoja, sekä mahdollisuus tallentaa tiketin numero robotin muistiin.

Työ kehitettiin Node.js ohjelmointikielellä ja työ on avoimen lähdekoodin ohjelmistoprojekti MIT-lisenssillä. Git oli tässä työssä käytetty versionhallintatyökalu ja sovelluksen lähdekoodi löytyy GitHubista.

### 1.1 Toteutustavan valinta

Hubot – ohjelmistorobotin varsinaisena kehityskielenä käytetään CoffeeScript ohjelmointikieltä, mikä on JavaScriptille kääntyvä ohjelmointikieli. Hubotille voi kuitenkin kirjoittaa skriptejä myös Node.js ohjelmointikieltä ja juuri Node.js valikoitui tämän työn ohjelmointikieleksi, koska se oli osin jo tuttu kieli ja se on hieman helpommin luettavaa kuin CoffeeScript.

Kaikki pyynnöt Hubotin ja ServiceNow:n välillä tässä työssä hoidettiin SOAP API rajapinnan kautta. ServiceNow tukee myös REST API tyyppisiä rajapintoja, mutta tässä työssä käytettiin servicenow-lite nimistä npm-pakettia, missä on valmiina funktion SOAP API:n käsittelyyn. On olemassa paljon muitakin npm paketteja ja luokkakirjastoja ServiceNow:n käsittelyyn eri rajapintojen kautta, mutta Servicenow-lite valittiin, koska sitä oli jo testattu aiemmin ja todettu toimivaksi. SOAP API:n käsittelyyn olisi myös voinut kirjoittaa oman luokkakirjaston, mutta määrittelyvaiheessa todettiin, että on helpompi käyttää valmiita kirjastoja ja siten vähennettiin kehitystyöhön liittyvää aikaa ja pystyi keskittymään paremmin pelkästään Hubotin ominaisuuksien kehittämiseen.

Hubot on mahdollista ottaa käyttöön monissa eri chatsovelluksissa, mutta tässä työssä Hubot kehitettiin Slackiin, koska se on käytössä Enfollla ja se tukee hyvin erilaisia kolmannen osapuolen ohjelmistoja ja niitä on helppo asentaa Slackiin.

Tekstieditorina lähdekoodin kirjoittamiseen käytettiin Visual Studio Code ohjelmaa, koska se oli jo ennestään tuttu ohjelmisto ja se tarjoaa tällä hetkellä laajan tuen erilaisille ohjelmistokielifille ja tekniikoille.

## 1.2 Toimeksiantaja

Enfo oyj on pohjoismainen, datavetoista liiketoimintaa mahdollistava IT-palvelutalo. Enfo hallitsee hybridialustat, informaatiohallinnan ja sovelluksien monitahoiset kokonaisuudet. He myös kehittävät erilaisia tapoja hallita datavirtaa, sekä jalostavat, analysoivat ja käyttävät tietoja. (Enfo Oyj, 2019)

Enfo Oyj:ssa työskentelee noin 900 työntekijää ja se on perustettu Kuopiossa vuonna 1964 alun perin nimellä Tietosavo oy. (Wikipedia, Wikipedia)



## 2 KEHITYSKOhteet JA TYÖKALUT

Tässä luvussa käsitellään työhön liittyviä kohteita ja ohjelmistorobottia, mille työ kehitettiin, sekä työhön liittyviä kehitystyökaluja ja määritelmiä.

### 2.1 Hubot

Hubot on GitHub oy:n kehittämä ohjelmistorobotti, joka kehitettiin alun perin automatisoimaan heidän yrityksensä chat – huonetta. Nykyään Hubot on avoimen lähdekoodin ohjelmisto, mikä on kirjoitettu CoffeeScript kielellä Node.js:lle. Se voidaan myös ottaa helposti käyttöön alustoilla, kuten Herokulla. Hubot on standardisoitu keino jakaa skriptejä eri Hubottien välillä. Hubot on mahdollista yhdistää erilaisiin chatsovelluksiin adaptereiden avulla. Adapteri toimii rajapintana palvelussa, missä Hubotin haluaa toimivan. Hubotilla on kaksi virallista adapteria Shellille ja Campfirelle, mutta myös monen kolmannen osapuolien adaptereita on kehitetty useimmille chatsovelluksille. Chatsovelluksia, mille adapteri löytyy, ovat mm. Gitter, HipChat, IRC ja Slack. (GitHub, Hubot)

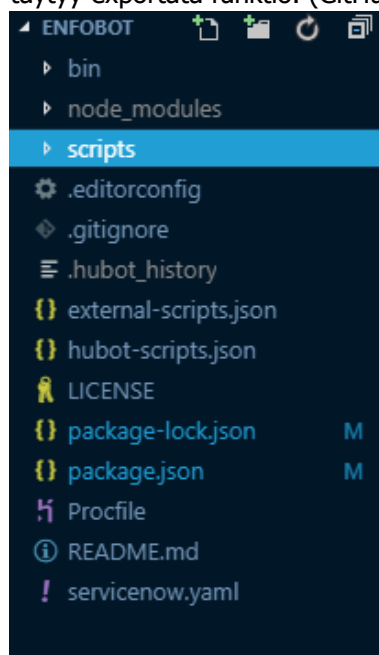
Hubotin mukana tulee sitä asennettaessa pieni joukko ydinskriptejä, jotka liittyvät esimerkiksi kielten käänkösiin kuvien julkaisuun ja Google Mapsin toimintoihin. Osia näistä toiminnoista on kuitenkin rajoitettu tätä raporttia kirjoittaessa. (GitHub, Hubot)

Ideana Hubotissa kuitenkin on, että siihen lisätään omia skriptejä yrityksen tarpeiden mukaan. Hubotille löytyy satoja muiden kirjoittamia valmiita skriptejä ja paketteja, mutta myös kenen tahansa kehittäjän on helppo alkaa kirjoittamaan omia skriptejään Hubotille.

#### 2.1.1 Hubotin skriptit

Hubotin skriptit sijaitsevat scripts – kansiossa, missä on luodessa valmiiksi yksi esimerkki skripti.

Skriptillä täytyy olla seuraavat ominaisuudet: niiden täytyy sijaita kansiossa, mikä on Hubotin skriptien latauspolussa src/scripts, skriptien täytyy olla CoffeeScript tai JavaScript tiedostoja ja skriptin täytyy exportata funktio. (GitHub, Scripting | Hubot)



KUVA 1. Hubotin kansiorakenne, mistä näkyy scripts kansion sijainti

```
module.exports = function (robot) {
  robot.commands.push(
    "Enfobot search <task number> - Prints info about task"
  );
};
```

KUVA 2. Esimerkki exportatusta funktiosta. Parametri robot kuvaa Hubotin instanssia.

Hubotin yleisimmät vuorovaikutukset perustuvat viesteihin, sillä Hubot on chatbot. Hubot kykenee kuuntelemaan viestejä tai vastaamaan viesteihin, joita sille annetaan suoraan. Molemmat metodit käyttävät säännöllisiä lausekkeita ja callback funktiota parametreinä. (GitHub, Scripting | Hubot)

```
robot.respond(/assign (.*) to person (.*)/i, function(response) {
  var ticketNumber = response.match[1];
  var person = response.match[2];
  var updateParams;

  console.log("Ticket number= " + ticketNumber);
  console.log("Person= " + person);
});
```

KUVA 3. Esimerkki viestistä mihin robotti vastaa

Metodin ollessa respond -tyyppinen, Hubot vastaa niihin robotin nimellä. Tämän jälkeen komentoa vastaava metodi triggeröityy. Yllä olevan esimerkin mukaan robotille sanottaisiin, että " (robotin nimi) assign task to person (henkilö)".

Robotti käyttää response parametria lukeakseen käyttäjän syöttämää tekstiä. Käyttäjän tekstisyöntein saamiseksi robotti käyttää JavaScriptin string match metodia, mikä etsii stringin vastaavuuksia säännölliselle lausekkeelle ja palauttaa vastaavuudet taulukko-objektina. (w3schools.com). Ylläolevasaa esimerkissä response.match[0] tarkoittaisi koko lausetta, response.match[1] ottaa ensimmäisen käyttäjän syöttämän sanan ja response.match[2] ottaa toisen käyttäjän syöttämän sanan ja nämä voidaan tallettaa muuttujiin.

Response parametrin avulla robotti myös vastaa käyttäjälle takaisin. Parametrin avulla voi käyttäjälle tulostaa esimerkiksi viestin toiminnon onnistumisesta - tai epäonnistumisesta.

```
response.send("Task " + ticketNumber + " assigned to " + person);
```

KUVA 4. Esimerkki robotin vastauksesta, mikä näkyy käyttäjälle takaisin

## 2.2 Git & GitHub

Git on Linus Torvaldsin vuonna 2005 kehittämä versiohallintatyökalu. Git on nykypäivänä käytössä todella monella avoimen lähdekoodin projekteissa, kuten myös kaupallisissa projekteissa ja se toimii monen eri käyttöjärjestelmän kanssa. Git käyttää jaettua DVCS (Distributed Version Control System) arkkitehtuuria, eli jokaisen kehittäjän luoma kopio koodista on myös säilytyspaikka, mikä sisältää täyden versiohistorian sovelluksesta. Git on suunniteltu turvalliseksi ja joustavaksi käyttää. (Bitbucket)

GitHub on vuonna 2008 perustettu yritys, mikä tarjoaa pilvipohjaisen säilytyspaikan ohjelmistojen lähdekoodeille. Pääasiallisesti GitHub tekee yksityisille kehittäjille ja kehitystiimeille versionhallinnan ja yhteistyön helpommaksi. GitHubin käyttöliittymä on käyttäjäystävällinen, joten Gitin käyttö aloittelijoillekin on helppoa. Ilman Githubia, Gitin käyttö vaatii enemmän teknistä osaamista. Kuka tahansa voi luoda GitHubiin tunnukset ja luoda repositoryn lähdekoodeille ilmaiseksi. Tämän takia GitHub on erityisen suosittu avoimen lähdekoodin projektien kanssa. (Kinsta, 2019)

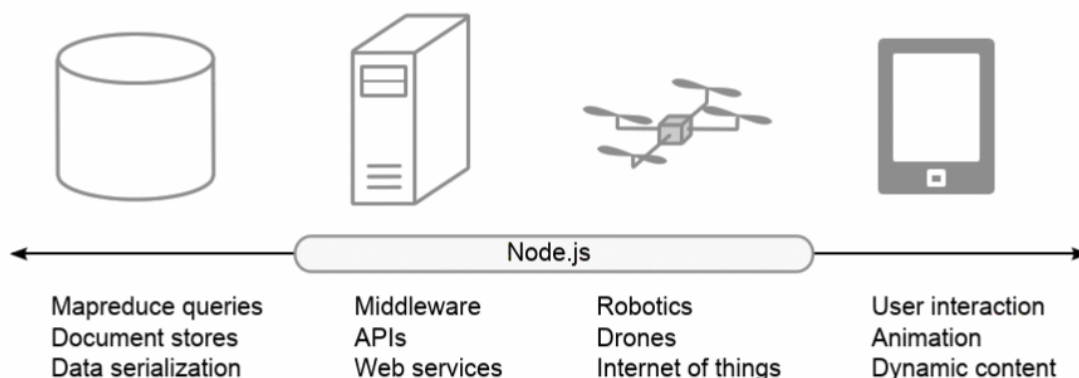
## 2.3 MIT – lisenssi

MIT (Massachusetts Institute of Technology) lisenssi on ilmainen lisenssi ohjelmistoille. Se mahdollistaa oikeudet loppukäyttäjälle muokkaamaan, kopioimaan, yhdistämään ja jakaa ohjelmistoa vapaasti. Lisenssi ei kuitenkaan sisällä ehtoja mainostukseen ja kieltoja käyttää omistajan nimen käyttöä oikeutta kaupalliseen käyttöön. (Technopedia)

## 2.4 Node.js

Node.js on alustariippumaton, avoimen lähdekoodin JavaScript runtime-ympäristö, jolla voidaan suorittaa JavaScript koodia, selaimen ulkopuolella. (Wikipedia) Node.js on suunniteltu skaalautuvia tietoverkkosovelluksia ja Node.js mahdollistaa palvelimen pään skriptien suorittamisen. (Foundation)

Suurin osa Node.js-sovelluksista on keskittynyt web-sovelluksiin. Kuitenkin Node.js:llä pystyy tekemään paljon muunlaisia sovelluksia kuin vain web-sovelluksia.



KUVA 5. Kuva sovellusten kirjosta, mitä Node.js kielellä pystyy kehittämään (Wilson, 2018).

Node.js on parhaiten suunniteltu palvelimenpään ohjelmiin, jotka odottavat asioita kuten tietokantayhteyttä, kolmannen osapuolen ohjelman palautetta tai tulevia yhteyspyyntöjä. (Wilson, Node.js 8 the Right Way : Practical, Server-Side JavaScript That Scales, 2018).

#### 2.4.1 Node.js:n toiminta

Node.js yhdistää JavaScriptin tapahtumasilmukkaan operaatioiden nopeasti lähettämiseksi tapahtuman ilmaantuessa. Tapahtumasilmukka on Node.js:n ydinominaisuus, mutta myös monet JavaScript ympäristöt käyttävät sitä. (Wilson, Node.js 8 the Right Way : Practical, Server-Side JavaScript That Scales, 2018)

Node.js antaa matalan tason pääsyn tapahtumasilmukkaan ja systeemin resursseihin. Niin kauan, kuin on jotain suoritettavaa, Node.js:n tapahtuma silmukka pyörii. Tapahtuman ilmaantuessa Node.js kutsuu callback-funktioita, mitkä kuuntelevat tapahtumaa. (Wilson, Node.js 8 the Right Way : Practical, Server-Side JavaScript That Scales, 2018)

Node.js:ää kehittäessä luodaan callback-funktioita, jotka suoritetaan vastauksena tapahtumiin. Callback-funktioita, jotka vastaavat kutsuihin, voi olla monia, mutta vain yhtä callback-funktiota voidaan suorittaa kerrallaan. Node.js suorittaa ohjelmiston koodia rinnakkaisesti. Mitään koodia ei suoriteta samaan aikaan kuin muuta koodia. (Wilson, Node.js 8 the Right Way : Practical, Server-Side JavaScript That Scales, 2018)

#### 2.5 ServiceNow

ServiceNow on pilvipohjainen IT Service Management (ITSM) alusta, mikä tukee Incident (tapaukset, tapahtumat), Problem (ongelmat), Change (muutokset) ja Knowledge (tieto) hallintaa ja monia muita IT prosesseja. Incident – hallinnan tarkoituksena on palauttaa normaali palvelu operaatio minimoiden vaikutukset liiketoimintaan. Problem – hallinta auttaa tunnistamaan virheiden syyt IT-infrastruktuurissa. Change – hallinta tarjoaa systemaattisen lähestymistavan hallitsemaan kaikkia muutoksia, mitä tapahtuu IT – palveluissa. Knowledge – hallinta sallii tiedon jakamisen knowledge base nimiseen palveluun. Knowledge base tarjoaa artikkeleita, missä esimerkiksi opastetaan käyttämään yrityksen sisäisiä palveluita. ServiceNow tarjoaa kattavan ITSM alustan ja sitä pystyy muokkaamaan laajasti eri yritysten tarpeisiin. ServiceNow:ssa on Information Technology Infrastructure Library eli ITIL sisällä, mikä mahdollistaa työnkulun tapaiset prosessit ja vahvan script-koneiston sekä palvelimenpään ja käyttäjänpään muokkaukseen. (Woodruff, 2017)

### 2.5.1 ServiceNow:n Taskit

Task taulu on ServiceNow:n taulu, mikä pitää yllä tietoja tehdyistä tehtävistä. Task – taulussa on lukuisia kenttiä, mitkä viittaavat työhön, mitä niille on tehty. Task – taulut sisältävät tiettyjä task kohtaisia ominaisuuksia, kuten work notes ominaisuuden, eli taskin kommentit, workflow integraation. Workflow:n avulla automatisoidaan ihmisen ja ohjelmiston välisiä prosesseja. Task – taulut sisältävät myös SLA (Service Level Agreement):n käsittelyn ja activity login, mikä näyttää kaikki toimet, mitä taskeille on tehty. (Woodruff, 2017)

Jokaisella Taskilla on ainakin seuraavat kentät:

- Active: Check-box kenttä, joka on joko true tai false. Yleensä jonkun business rulen kontrolloima
- Additional comments and work notes: Kommentti kenttä: Additional comments, kenttää voivat muokata ketkä tahansa käyttäjät, mutta work notes kenttää vain ITIL roolin omaavat käyttäjät. ITIL (Information Technology Infrastructure Library) on kokoelma käytäntöjä, milä johdetaan ja hallitaan IT-palveluita. ITIL-rooleilla määritetään vastuita erilaisille ITIL prosesseille.
- Assigned to ja assignment group: Viittaa käyttäjät tauluun ja kertoo kelle käyttäjälle tai ryhmälle task on osoitettu.
- Created, created by: Kenttä, mikä kertoo, kuka on taskin luonut.
- Updated ja updated by: Kertoo, kuka on päivittänyt taskia
- Short description ja description: Tarkoittaa lyhyttä - ja pitempää kuvausta taskista.
- Number: Identifioi taskit. Esimerkiksi incident taskilla number on INC123456.
- State: Ilmoittaa taskin tilan. Tiloja ovat esimerkiksi new, active, resolved, closed
- Journals: Ei varsinaisesti tallenna dataa tauluun, missä Journal kenttä on, vaan luo kirjauksia Journal Entries tauluun. Journal kentät sisältävät kentät kommentteista, mitä taskille on annettu.

### 2.6 Slack

Slack on pilvipohjainen yhteistyöalusta tiimeille ja sitä käytetään lähinnä yritysten ja organisaatioiden sisäisessä viestinnässä. Slack tarjoaa erilaisia IRC-tyyppisiä ominaisuuksia, kuten aiheen mukaan järjestettyjä chat huoneita, yksityisiä ryhmiä sekä suoria viestiä. Slackin kautta pystyy myös lähettämään tiedostoja ja sen avulla pystyy käymään video -tai puhelinneuvotteluja. (Slack)

Slackiin on mahdollista integroida kolmannen osapuolen tarjoamia ohjelmistoja ja käyttää niitä suoraan Slackissa. Slackiin on mahdollista saada yli 1000 erilaisia integroitua ohjelmistoja, joista suosituimpia ovat mm. Google Drive, Dropbox, Heroku ja GitHub. Hyöty integroiduista ohjelmista on, että Slackia käyttävän henkilön ei tarvitse vaihtaa välilehtiä tai avata ohjelmaa vain tarkistaakseen nopeasti jonkun asian. (CloudApp: Emilie Johnston, 2019)

## 2.7 SOAP API

SOAP (Simple Object Access Protocol) on XML pohjainen vahvasti tyypitetty protokolla, millä on mahdollista päästä web-palveluihin http:n yli. SOAP pohjaiset rajapinnat on suunniteltu siten, että niillä pystyy luomaan, muokkaamaan, hakemaan ja poistamaan objekteja. Jokainen toiminto on selvästi määritetty XML rakenteen mukaan, mutta suurin osa SOAP sovelluksista käyttävät WSDL (Web Services Definition Language) kieltä. (Techtarget)

## 2.8 Visual Studio Code

Visual Studio Code on Microsoftin kehittämä avoimen lähdekoodin ohjelmoijille tarkoitettu tekstieditori, joka on tarjolla Microsoft, Linux ja MacOS käyttöjärjestelmille. Visual Studio Codessa on sisäänrakennettu tuki erilaisille ohjelmointikielille, kuten JavaScriptille, Node.js:lle ja TypeScriptille ja siihen saa laajennuksia eri kielille, kuten Pythonille, PHP:lle ja Javalle. Laajennuksia on myös tarjolla Azurelle, Gitille ja Dockerille. (Code)

### 3 KEHITYSTYÖN VAIHEET

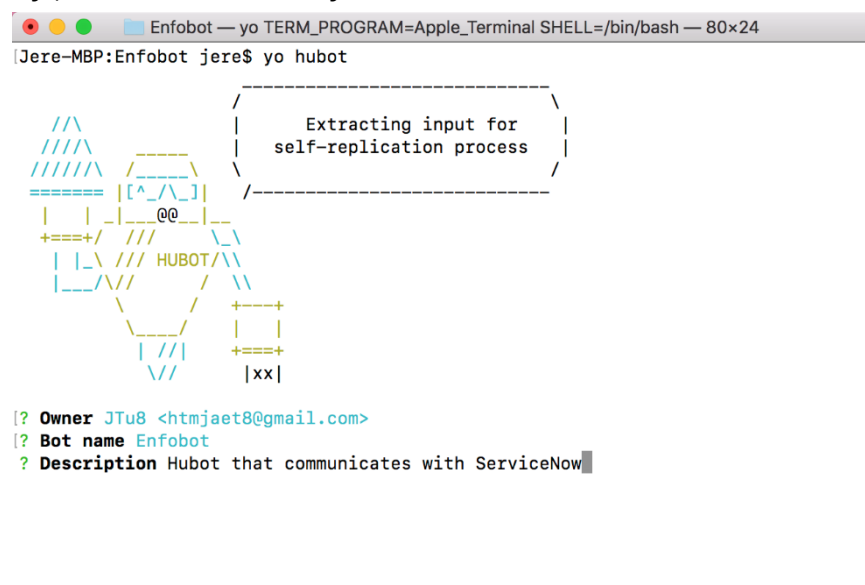
#### 3.1 Hubotin asennus

Sovellus aloitettiin Hubotin asennuksella. Ennen Hubotin asentamista täytyi varmistaa, että Node.js ja npm ovat asennettuina. Tämän jälkeen asennettiin Hubot Yeoman generaattorin avulla. Yeoman generaattori on avoimen lähdekoodin rakennustyökalu moderneille web – sovelluksille, jonka avulla voi rakentaa malleja erilaisista sovelluksista (Yeoman, Yeoman). Jokainen generaattori toimii laajenuksena Yeoman ympäristössä. (Yeoman, Yeoman) Kun Yeoman generaattori oli asennettu, asennettiin Hubot ”yo hubot” – komennolla.



KUVA 6. Hubotin asennus aloitettu

Komennon antamisen jälkeen asennusvaiheessa piti antaa tietoja esimerkiksi, kuka on Hubotin omistaja, mikä on Hubotin nimi ja kuvaus Hubotin toiminnosta.



KUVA 7. Asetetaan Hubotin vaatimat tiedot

Tietojen antamisen jälkeen Hubot asentui ja tämän jälkeen Hubot käynnistyy kirjoittamalla komentorivillä "bin\hubot" – komennon, mikä käynnistää Hubotin shell adapterilla. Shell adapteria käytettiin alussa kehitystä varten ja myöhemmin siirryttiin käyttämään Slackin adapteria.

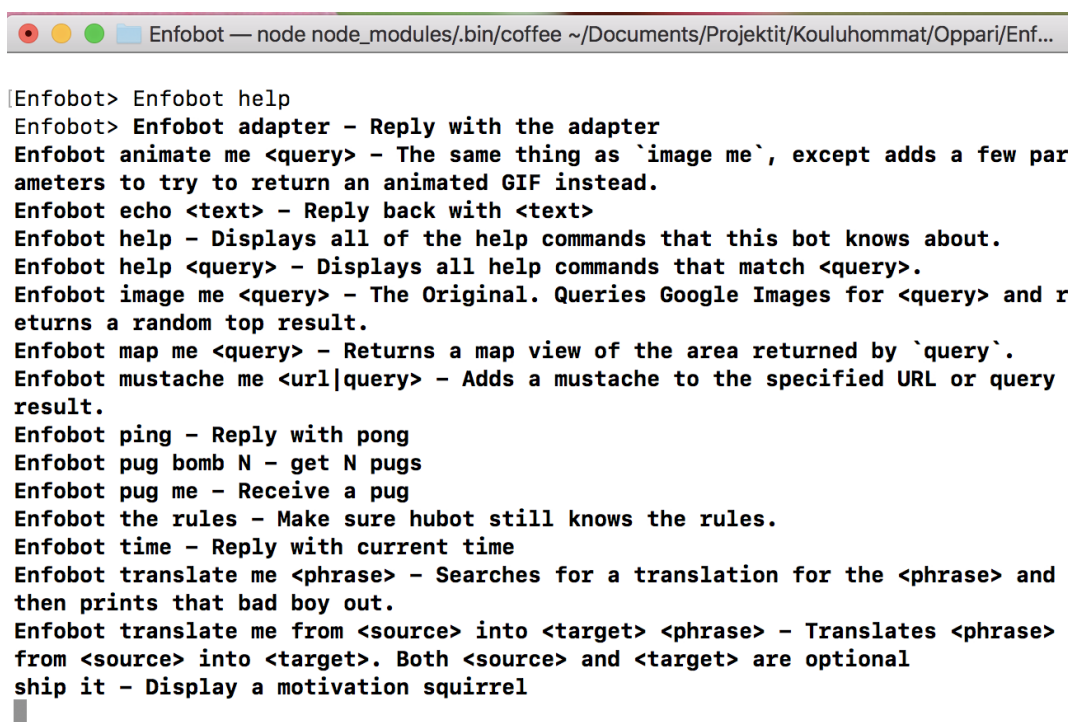
```
No history available
Enfobot> [Tue Apr 30 2019 09:36:01 GMT+0300 (GMT+03:00)] WARNING Loading scripts
  from hubot-scripts.json is deprecated and will be removed in 3.0 (https://github
  b.com/github/hubot-scripts/issues/1113) in favor of packages for each script.

Your hubot-scripts.json is empty, so you just need to remove it.
[Tue Apr 30 2019 09:36:01 GMT+0300 (GMT+03:00)] INFO hubot-redis-brain: Using de
fault redis on localhost:6379
[Tue Apr 30 2019 09:36:01 GMT+0300 (GMT+03:00)] ERROR hubot-heroku-keepalive inc
luded, but missing HUBOT_HEROKU_KEEPALIVE_URL. `heroku config:set HUBOT_HEROKU_K
EEPALIVE_URL=$(heroku apps:info -s | grep web.url | cut -d= -f2)`

Enfobot> █
```

KUVA 8. bin\hubot komento ajettu shell adapteri käynnistetty

Hubotin ollessa käynnissä ensimmäisenä täytyi testata, että Hubot vastaa komentoihin kirjoittamalla komentoriville "help" – komennon, mikä käynnistää Hubotin listan olemassa olevista Hubotin komennoista.



```
[Enfobot> Enfobot help ]
Enfobot> Enfobot adapter – Reply with the adapter
Enfobot animate me <query> – The same thing as `image me`, except adds a few par
ameters to try to return an animated GIF instead.
Enfobot echo <text> – Reply back with <text>
Enfobot help – Displays all of the help commands that this bot knows about.
Enfobot help <query> – Displays all help commands that match <query>.
Enfobot image me <query> – The Original. Queries Google Images for <query> and r
eturns a random top result.
Enfobot map me <query> – Returns a map view of the area returned by `query`.
Enfobot mustache me <url|query> – Adds a mustache to the specified URL or query
result.
Enfobot ping – Reply with pong
Enfobot pug bomb N – get N pugs
Enfobot pug me – Receive a pug
Enfobot the rules – Make sure hubot still knows the rules.
Enfobot time – Reply with current time
Enfobot translate me <phrase> – Searches for a translation for the <phrase> and
then prints that bad boy out.
Enfobot translate me from <source> into <target> <phrase> – Translates <phrase>
from <source> into <target>. Both <source> and <target> are optional
ship it – Display a motivation squirrel
█
```

KUVA 9. Hubotin komennot listattuna



### 3.2 Slack adapterin käyttöönotto

Kehitystyötä varten luotiin oma Slack-kanava, jossa Hubotin ominaisuuksia testattiin. Jotta Hubot saatiin toimimaan Slackissa, täytyi se ensiksi asentaa Slackin App Directory sivun kautta, mistä löytyy paljon kolmannen osapuolen sovelluksia Slackiin. Asennuksen jälkeen saadaan Hubot-slack-token, mitä käytetään autentikoimaan Hubot ja se annetaan komentorivillä ennen `bin\hubot` komentoa, jolloin Hubot alkaa käyttämään Slackin adapteria.

### 3.3 Ominaisuuksien luominen

Hubotin toiminnan testaamisen jälkeen oli seuraava työvaihe aloittaa omien skriptien lisääminen ja suunnitelmissa laadittujen ominaisuuksien luominen. Ennen työn aloittamista suunnittelupalaverissa asetettiin tavoitteet ja ominaisuudet, mitkä Hubotille kehitetään. Tässä työssä käytettiin hyväksi `servicenow-lite` nimistä npm-pakettia. `ServiceNow-lite` tarjoaa funktiot `ServiceNow:n` SOAP API:n käsittelyyn (npm). Osa paketissa olevista funktiosta oli jo valmiina ja niistä oli työssä paljon hyötyä, mutta esimerkiksi `ServiceNow:n` tikkettien päivittämiseen liittyvä funktio puuttui, joten se täytyi itse lisätä `servicenow-liten` lähdekoodiin. Suurin osa kehitettävistä ominaisuuksista liittyy jollain tavalla tikkettien kenttien muokkaamiseen. Tämän lisäksi ennen kuin alkaa käyttämään `ServiceNow:n` SOAP API:a hyödyksi, täytyy asettaa `ServiceNow:n` käyttäjän käyttäjätunnus ja salasana ympäristömuuttujina.

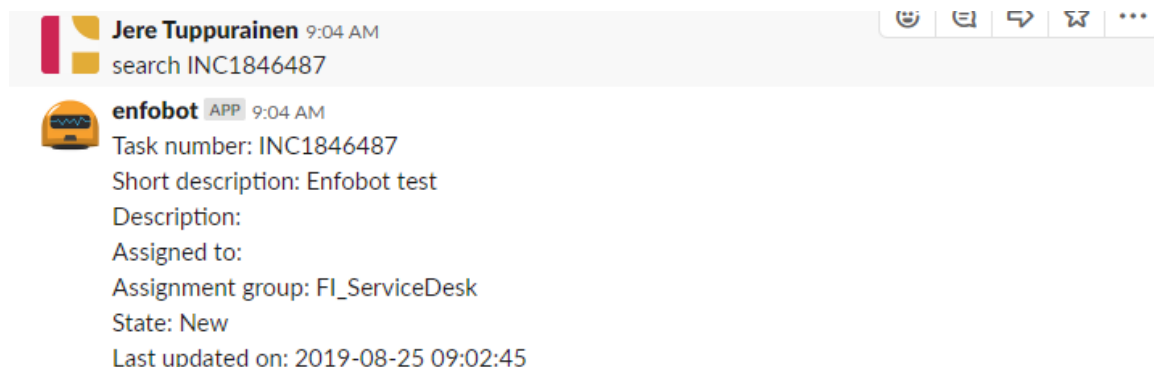
```
// Updates ticket by tickets number
function updateTicket(params, id, callback) {
  getClient(config.tableName(id), function(err, client) {
    if (err) {
      callback(err, null);
      return;
    }
    client.update(params, callback);
  });
}
```

KUVA 10. Update funktio, mikä on itse lisätty `servicenow-liten` `index.js` tiedostoon

Projektin juurikansioon täytyy myös laittaa `yaml` -tiedosto, minne laitetaan url – osoite käytettävän `ServiceNow:n` instanssiin, sekä tarvittavat `ServiceNow:n` taulut, mitä Hubotin halutaan käsittelevän.

`Yaml` on data orientoitunut kielirakenne, mitä käytetään syöteformaattina erilaisissa ohjelmistosovelluksissa. Sitä ei ole tarkoitettu merkintäkieleksi. `Yaml`-tiedostoa käytetään esimerkiksi määrittämään käyttäjät, tai esimerkiksi asennettavat ohjelmistopakettit. (`TechTarget`)

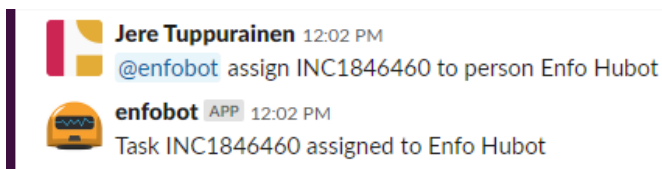
Ennen varsinaisia pääominaisuuksien luonteja luotiin ominaisuus, millä voi hakea tietoja tiketistä. Ominaisuus luotiin Hubotin ja ServiceNow:n välisen yhteyden testaamiseksi. Tietojen haun yhteydessä tulostetaan myös linkki tiketille, mikä avaa selaimen suoraan tiketille.



KUVA 11. Tiketin tietojen haku. Kuvassa näkyy kentät mitä tiketiltä haetaan.

### 3.3.1 Tiketin ohjaaminen

Tämän jälkeen kehitettävä osio oli tiketin ohjaus toiselle henkilölle tai ryhmälle. Hubot ottaa parametreina käyttäjältä tiketin numeron ja henkilön tai ryhmän nimen kenelle tiketti ohjataan. Tämän jälkeen Hubot tarkistaa käyttäjä taulusta (sys\_user), että onko sen nimistä käyttäjää olemassa ja tarkastaa myös käyttäjän sys\_id:n ja yrityksen, että tiketti menee varmasti oikealle käyttäjälle, jos käyttäjä löytyy Hubot päivittää tiketiltä assigned to kentän ja ilmoittaa käyttäjälle, että toiminto onnistui tai epäonnistui. Assignment group kenttää varten Hubot tarkistaa sys\_group taulusta, että löytyykö vastaavaa ryhmää.



KUVA 12. Tiketti ohjataan henkilölle

Yllä olevasta kuvasta näkyy tiketin ohjaus henkilölle ja botin vastaus käyttäjälle, kun toiminto onnistuu. Jos tiketti halutaan ohjata ryhmälle, komentoon kirjoitetaan to group.

### 3.3.2 Tiketin kommentointi

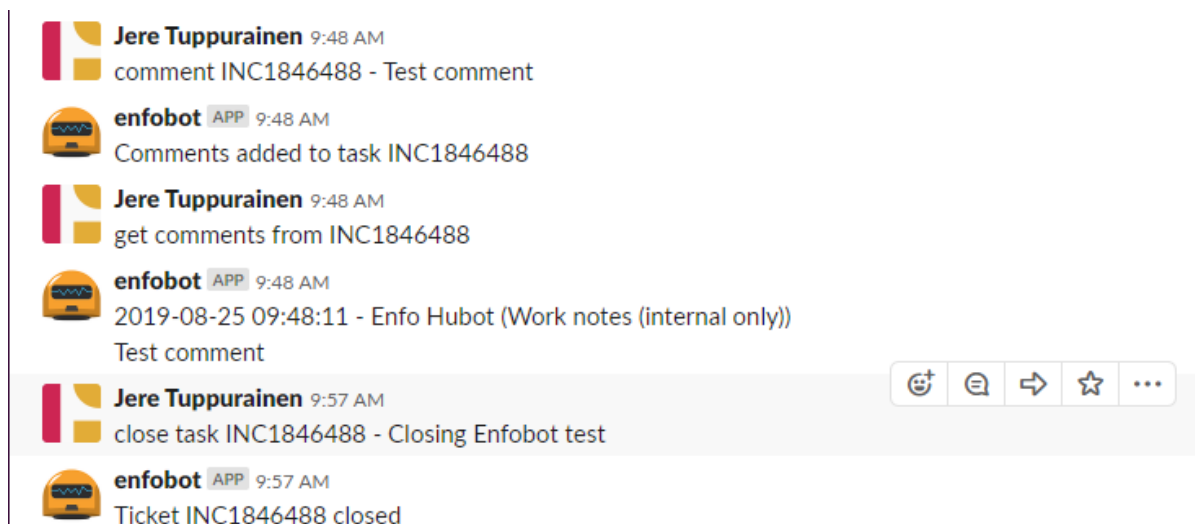
Toinen kehitettävä ominaisuus oli tiketin kommentointi. Kommentointi tapahtuu antamalla Hubotille tiketin numeron, mille haluaa kommentin antaa ja itse kommentin sisällön. Hubot ilmoittaa tämän jälkeen, mitä tikettiä oli kommentoitu tai ilmoittaa jos toiminto epäonnistui. Tämä toiminto päivittää tiketin work notes (internal) kenttää eli kommentit näkyvät vain sisäisesti eivätkä esimerkiksi asiakkaille.

### 3.3.3 Tietin sulkeminen

Kolmas ominaisuus oli tietin sulkeminen. Tietin sulkeminen tapahtuu antamalla Hubotille tietin numero ja tietin sulkemiseen tarvittavat kommentit. Tietin sulkemisen onnistuessa Hubot antaa siitä ilmoituksen käyttäjälle ja myös viestin, jos tapahtuma epäonnistuu. Tämä toiminto päivittää tietiltä state kenttää ja tässä piti ottaa huomioon, minkälainen tietti oli kyseessä. Esimerkiksi incident tyyppistä tiettiä ei voida suoraan sulkea closed tilaan, vaan sen pitää ensiksi olla resolved tilassa vähintään 5 päivää, minkä jälkeen se automaattisesti menee closed tilaan.

### 3.3.4 Tietin kommenttien näyttäminen

Neljäs kehitettävä ominaisuus oli tietin kommenttien näyttäminen. Kommenttien näyttäminen tapahtuu antamalla Hubotille komento ja tietin numero, minkä tietin kommentit haluaa nähdä. Hubot tarkastaa löytyykö annettua tiettiä ja jos löytyy, Hubot tulostaa tietin kaikki annetut kommentit järjestyksessä viimeisin kommentti ensimmäisenä.



KUVA 13. Kommentointi, kommenttien hakeminen ja tietin sulkeminen

### 3.3.5 Tiketin talletus muistiin

Näiden ominaisuuksien lisäksi käyttäjälle laitettiin myös mahdollisuus tallettaa tiketin numero ja oma käyttäjänimensä Hubotin muistiin. Tämä toiminto toteutettiin, koska tämän avulla kaikki edellä mainitut ominaisuudet toimivat ilman, että tiketin numeroa tarvitsee koko ajan syöttää Hubotille, vaan Hubot lukee tiketin numeron muistista.

Hubotilla on avain arvo pareihin perustuva datan tallennus, mitä kutsutaan käyttämällä `robot.brain` metodia.













```
// Adds task to memory
robot.respond(/save task (.*)/i, function(response) {
  var task = response.match[1];

  robot.brain.set('task', task);
  response.send("Task number " + task + " has been saved");
});
```

KUVA 14. Script millä tiketti talletetaan Hubotin muistiin

Yllä olevassa kuvassa tiketti talletetaan Hubotin muistiin `robot.brain.set` metodilla. Tiketti saadaan haettua muistista `robot.brain.get` metodilla.

Slackissa käyttäjän halutessa käsitellä tikettiä Hubotin muistin kautta, käyttäjän täytyy antaa Hubotille komento `activate` (tiketin numero), jolloin Hubot tallettaa tiketin numeron muistiin ja tulostaa talletetusta tiketistä tiedot sekä komennot, mitä voi käyttää, kun tiketti on talletettuna muistiin. Hubotin muistissa voi olla vain yksi tiketti kerrallaan. Komentoja antaessa on myös tehty tarkistukset, että onko tiketti varmasti Hubotin muistissa. Tiketin ollessa käsitelty, annetaan Hubotille komento `forget`, jolloin tiketti pyyhitään pois muistista.

-  **Jere Tuppurainen** 9:03 AM  
active INC1846490
-  **enfobot** APP 9:03 AM  
INC1846490 is now active
- 9:03 AM Short description: Enfobot test  
Description:  
Assigned to:  
Assignment group: FI\_ServiceDesk  
State: New  
Last updated on: 2019-09-01 09:02:58  
Link: <https://enfo.com/.../enfo-hubot-test-de2019-09-01-09-02-58-1846490/>
- Typical commands: assign to me, give comment <comments>, show comments, close active task <close notes>
-  **Jere Tuppurainen** 9:03 AM  
assign to me
-  **enfobot** APP 9:03 AM  
Task INC1846490 assigned to Jere Tuppurainen
-  **Jere Tuppurainen** 9:03 AM  
give comment Test comment
-  **enfobot** APP 9:03 AM  
Comments added to task INC1846490
-  **Jere Tuppurainen** 9:03 AM  
show comments
-  **enfobot** APP 9:04 AM  
2019-09-01 09:03:43 - Enfo Hubot (Work notes (internal only))  
Test comment
-  **Jere Tuppurainen** 9:04 AM  
close active task Closing test
-  **enfobot** APP 9:04 AM  
Ticket INC1846490 closed
-  **Jere Tuppurainen** 9:04 AM  
forget
-  **enfobot** APP 9:04 AM  
No active task number.

KUVA 15 Tiketin tallennus muistiin ja sen käsittely

## 4 LOPPUTULOS

Työn lopputuloksena syntyi Slackissa toimiva ohjelmistorobotti, millä pystyy käsittelemään Service-Now:n tikettejä. Työhön kehitettiin seuraavanlaiset ominaisuudet:

- Tiketin ohjaus toiselle henkilölle
- Tiketin tietojen hakeminen
- Tiketin kommentointi
- Tiketin kommenttien hakeminen
- Tiketin käsittely muistista

Työ oli avoimen lähdekoodin projekti ja sille laitettiin MIT:n lisenssi. Loppuvaiheessa minun kehittämät ominaisuudet yhdistettiin Enfollla olevan Hubotin kanssa, mikä pyörii Kubernetes ympäristössä. Tätä yhdistämisvaihetta en itse tehnyt, mutta olin seuraamassa, kun se toteutettiin.

Opinnäytetyön kehitysvaihe sujui mielestäni suhteellisen hyvin. Olin Enfollla harjoittelussa kolmen kuukauden ajan ja sen lopulla pääsin jo tutustumaan Hubot-ohjelmistorobottiin ja sen skripteihin, joten sen kehitys oli jo osin tuttua ja vähensi uusien asioiden opettelua opinnäytetyötä tehdessä.

Opinnäytetyön kehitys eteni mielestäni hyvin ja pysyin määritellyssä aikataulussa koko sen kehityksen ajan. Ainoat kehitystyön aikana tulleet haasteet liittyivät ServiceNow:n tauluihin ja tikettien erilaisten kenttien määrittelyihin, mutta nekin sai hoidettua joko kyselemällä tai itse selvittämällä. Hubotin skripteihin liittyviä haasteita tai ongelmia ei juurikaan ollut, vaan oivalsin aika nopeasti, miten Hubot-ohjelmistorobottia kehitetään.

Jatkokehityskohteita Hubotille olisi varmasti paljonkin, sillä jo määrittelyvaiheessa tuli esille paljon ominaisuuksia, mitä voisi kehittää. Näitä ominaisuuksia olivat mm. Knowledge artikkelien hakeminen ja CMDB tietojen etsintä. Kommenttien hakemiseen voisi myös kehittää ominaisuuden, jolla voi päätää miten monta kommenttia tiketiltä hakee kerrallaan.

## 5 LÄHDELUETTELO

- Bitbucket, A. (ei pvm). *Atlassian Bitbucket: Tutorials*. Haettu 24. Elokuu 2019 osoitteesta <https://www.atlassian.com/git/tutorials/what-is-git>
- CloudApp: Emilie Johnston. (14. Huhtikuu 2019). *Marketing & Sales: CloudApp*. Haettu 28. Heinäkuu 2019 osoitteesta <https://www.getcloudapp.com/blog/what-is-slack>
- Code, V. S. (ei pvm). *Getting Started*. Haettu 4. Elokuu 2019 osoitteesta <https://code.visualstudio.com/docs>
- Enfo Oyj. (2019). *Enfo*. Haettu 23. Kesäkuu 2019 osoitteesta <https://www.enfo.fi/meista>
- Foundation, N. (ei pvm). *Node.js*. Haettu 7. Heinäkuu 2019 osoitteesta <https://nodejs.org/en/about/>
- GitHub. (ei pvm). *Hubot*. Haettu 29. Kesäkuu 2019 osoitteesta <https://hubot.github.com/>
- GitHub. (ei pvm). *Scripting / Hubot*. Haettu 6. Heinäkuu 2019 osoitteesta <https://hubot.github.com/docs/scripting/>
- Kinsta. (22. Elokuu 2019). *Kinsta: What Is GitHub? A Beginner's Introduction to GitHub*. Haettu 24. Elokuu 2019 osoitteesta <https://kinsta.com/knowledgebase/what-is-github/>
- npm. (ei pvm). *servicenow-lite*. Haettu 4. Elokuu 2019 osoitteesta <https://www.npmjs.com/package/servicenow-lite>
- Slack. (ei pvm). *Help Center: Slack*. Haettu 28. Heinäkuu 2019 osoitteesta <https://get.slack.help/hc/en-us/articles/115004071768-What-is-Slack->
- Technopedia. (ei pvm). *Technopedia*. Haettu Elokuu 2019 osoitteesta <https://www.techopedia.com/definition/3287/mit-license>
- Techtarget. (ei pvm). Haettu 25. Elokuu 2019 osoitteesta <https://searcharchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol>
- TechTarget. (ei pvm). Haettu 25. Elokuu 2019 osoitteesta <https://searchitoperations.techtarget.com/definition/YAML-YAML-Aint-Markup-Language>
- w3schools.com. (ei pvm). *w3schools.com*. Haettu 7. Heinäkuu 2019 osoitteesta [https://www.w3schools.com/jsref/jsref\\_match.asp](https://www.w3schools.com/jsref/jsref_match.asp)
- Wikipedia. (ei pvm). *Wikipedia*. Haettu 23. Kesäkuu 2019 osoitteesta <https://fi.wikipedia.org/wiki/Enfo>
- Wikipedia. (ei pvm). *Wikipedia*. Haettu 7. Heinäkuu 2019 osoitteesta <https://en.wikipedia.org/wiki/Node.js>
- Wilson, J. (2018). *Node. js 8 the Right Way : Practical, Server-Side JavaScript That Scales*. Haettu 13. Heinäkuu 2019
- Wilson, J. (2018). *Node. js 8 the Right Way : Practical, Server-Side JavaScript That Scales*. Pragmatic Programmers, LLC, The. Haettu 13. Heinäkuu 2019
- Woodruff, T. (2017). *Learning ServiceNow* (Osa/vuosik. 1). Packt Publishing, Limited. Haettu 12. Elokuu 2019
- Yeoman. (ei pvm). *Yeoman*. Haettu 3. Elokuu 2019 osoitteesta <https://yeoman.io/>
- Yeoman. (ei pvm). *Yeoman*. Haettu 3. Elokuu 2019 osoitteesta <https://yeoman.io/learning/index.html>